UNIVERSIDAD DE LA RIOJA

TRABAJO FIN DE ESTUDIOS

Título

Beneficios en torno al aprendizaje de la programación

Autor/es

IRATI ARRONIZ NAZABAL

Director/es

JULIO BLANCO FERNÁNDEZ

Facultad

Escuela de Máster y Doctorado de la Universidad de La Rioja

Titulación

Máster Universitario de Profesorado, especialidad Tecnología

Departamento

INGENIERÍA MECÁNICA

Curso académico

2018-19

publicaciones.unirioja.es
E-mail: publicaciones@unirioja.es

Trabajo de Fin de Máster

Beneficios entorno al aprendizaje de la programación

Autora

Irati Arroniz Nazabal

Tutor: Julio Blanco Fernandez

MÁSTER:
Máster en Profesorado, Tecnología (M07A)

Escuela de Máster y Doctorado

AÑO ACADÉMICO:2018/2019

Sumario

1 RESUMEN.....5
2 INTRODUCCIÓN Y JUSTIFICACIÓN.....7
3 OBJETIVOS.....11
3.1 TIC.....11
3.2 Aprendizaje para la vida.....12
3.3 Aprender a través de las experiencias.....13
3.4 Pensamiento computacional.....14
3.5 Pensamiento lógico y algorítmico.....14
3.6 Pensamiento lógico matemático.....15
4. MARCO TEÓRICO.....17
4.1 Constructivismo.....17
4.2 Scratch.....17
4.2.1 Dr. Scratch.....19
4.3 Scratch como herramienta inclusiva.....19
4.3.1 Scratch herramienta para alumnos con necesidades educativas especiales (NEE).....20
4.3.2 Ejemplos Rampas Digitales.....21
4.4 Marco legal.....23
5. ESTADO DE LA CUESTIÓN.....25
5.1 Docentes.....25
5.2 Estudios consultados.....25
5.2.1 Habilidades desarrolladas mediante el aprendizaje de la programación.....25
5.2.2 Estudios entorno a Scratch, Dr. Scratch y Alumnos con NEE.....26
5.2.3 Definición y componentes de la autoestima.....26
5.2.4 Definición y construcción del autoconcepto.....27
6 METODOLOGÍA DE LOS ESTUDIOS.....29
7 RESULTADOS.....31
7.1 Estudios entorno a las habilidades desarrolladas mediante el aprendizaje de la programación.....31

7.1.1 Beneficios del desarrollo del pensamiento computacional y el pensamiento algorítmico.....32
7.2 Estudios entorno a Scratch, Dr. Scratch y alumnos con NEE.....33
8. CONCLUSIONES.....37
9. REFERENCIAS.....41

1 RESUMEN

A lo largo de esta investigación se examinan diferentes estudios que han analizado los beneficios que aporta el aprender a programar y desarrollar el pensamiento computacional y se extraen los resultados en los que coinciden la mayor parte de ellos. Estos estudios analizan la destreza matemática, la capacidad de concentración y las habilidades de pensamiento y razonamiento condicional, procedimental y temporal como aportaciones al desarrollo cognitivo mediante la programación informática . Así como, estudios que han utilizado la herramienta Scratch para el aprendizaje de la programación destacándola como recurso didáctico apropiado. Se analizan características como el fomento de la capacidad creativa o el desarrollo de estrategias para organizar la información y resolver problemas. Se aprecia un aumento de la motivación entorno al aprendizaje y un aumento de la capacidad colaborativa. También, se analiza la herramienta Dr. Scratch como instrumento potenciador del aprendizaje y la motivación. Así como, las adaptaciones posibles mediante Rampas Digitales o Assistive Technology para hacer accesible las diferentes herramientas a alumnos con necesidades educativas especiales. Se mencionan los resultados observados en alumnos con autismo y discapacidad intelectual. Por último, se relacionan las capacidades mencionadas con conceptos como la seguridad, la integración, la comunicación y la necesidad de estrategias de organización para analizar y reflexionar entorno a las experiencias, todas ellas capacidades relacionadas a la autoestima intelectual y el autoconcepto.

Por lo tanto, el objetivo de este análisis es relacionar las habilidades y capacidades que se desarrollan mediante el aprendizaje de la programación con las características que componen el autoconcepto y la autoestima. Con el fin de proponer un cuestionario para alumnos que hayan aprendido a programar con Scratch o una herramienta similar. Las preguntas del cuestionario giran entorno a conceptos como la motivación, la comunicación, la seguridad, la autonomía, la capacidad colaborativa o las estrategias para resolver problemas.

ABSTRACT

I have analyzed different studies that examine the benefits of learning to program and the benefits of developing computational thinking. They analyze mathematical skill, the capacity of concentration and conditional, procedural and temporal reasoning as contributions to cognitive development through computer programming. Also, studies that use Scratch as an appropriate educational resource for learning to program. These studies analyze different skills, for example, the development of creativity or the development of strategies to organize information and solve problems. They appreciate that motivation around learning and collaborative capacity have increased. Also, they analyze Dr. Scratch as a tool that enhances learning and motivation and tools as Digital Ramps or Assistive Technology that makes accessible the different instruments to students with special educational needs. I mention the results observed in students with autism and intellectual disability.

Finally, I have tried to connect the capacities that I have mentioned with concepts as security, integration, communication and the need for organizational strategies to analyze and reflect on our experiences. These concepts are related to self-esteem and self-concept. Therefore, the objective of this analysis is to connect the skills and abilities that are developed by learning to program with the skills that make up the self-concept and self-esteem. I propose a questionnaire for students who have learned to program with Scratch or a similar tool. These questions are related to motivation, communication, security, autonomy, collaborative capacity and strategies to solve problems.

Palabras clave: Beneficios, programación, pensamiento computacional, desarrollo cognitivo, Scratch, autoconcepto, autoestima, cuestionario.

Key words: Benefits, programming, computational thinking, cognitive development, Scratch, self-concept, self-esteem, questionnaire.

2 INTRODUCCIÓN Y JUSTIFICACIÓN

La sociedad hoy en día esta inmersa en la era de la información, el manejo y el intercambio de información a través de las TIC esta cobrando relevancia en diferentes aspectos de la sociedad. Las TIC están presentes en el ámbito educativo, en el ámbito profesional y en el ámbito social.

Hoy en día los ordenadores y las herramientas tecnológicas son parte de la vida de los niños pero pocos aprenden a programar, pocos aprenden sobre el funcionamiento y la lógica que siguen los aparatos tecnológicos. La gran mayoría de los alumnos son meros usuarios de estas herramientas tecnológicas. Las utilizan para comunicarse con sus amigos, jugar o ver videos, pero pocos son capaces de crear contenido. Es como si pudieran leer pero no escribir (Resnick, Maloney, Monroy-Hernández, Rusk, Eastmond, Brennan, Millner, Rosenbaum, Siver, Silverman, Kafay, 2009).

En consecuencia, las leyes entorno a la educación ya han sido adaptadas para incluir las TIC como parte del proceso educativo de los alumnos. También, se ve reflejado en diferentes estrategias políticas a nivel comunitario, estatal y autonómico. Por ejemplo, existen estrategias políticas como “Europa 2020” cuyo objetivo principal es la alfabetización en el entorno digital, remarcando como esta alfabetización va ser completamente necesaria en el mercado laboral del futuro. La nanotecnología o la robótica cobran gran importancia en el mercado laboral, por lo tanto, se requiere de personas con formación en informática para poder cubrir las necesidades actuales. En definitiva, se requiere personal que haya desarrollado el pensamiento computacional como una parte más de sus competencias.

A lo largo de este trabajo me gustaría remarcar como el hecho de aprender a programar beneficia el desarrollo de ciertos aspectos cognitivos, que a su vez, ayudan a los alumnos a disponer de un mayor número de herramientas estratégicas para poder resolver problemas del cotidiano y tener una mayor comprensión acerca de las herramientas tecnológicas con las que se relacionan constantemente.

Tal y como se ha mencionado, nos encontramos en la era de la información y la tecnología pero se puede afirmar que el simple hecho de disponer de información no genera conocimiento. Para que la información genere conocimiento se debe comprender e integrar en los esquemas del conocimiento previo que dispone cada persona. Para ello, se requieren estrategias de razonamiento para organizar, analizar, relacionar y sintetizar la información de la que se dispone.(Morrás, 2014). Esto supone operar en diferentes niveles de abstracción simultáneamente, definiendo, automatizando o mecanizando los diferentes niveles y sus relaciones.

El aprendizaje debe de estar relacionado con la asimilación de las actividades, de esta manera, el aprendizaje es parte de un proceso personal donde los recursos cognitivos de cada uno marcarán el resultado obtenido.

Diferentes autores y estudios afirman que ha mediados del siglo XXI el pensamiento computacional va ser una competencia tan importante como leer, escribir o la aritmética. Wing (2011)

A lo largo de este trabajo se relaciona el hecho de aprender a programar con el desarrollo del pensamiento computacional, el pensamiento lógico, el pensamiento matemático y el pensamiento algorítmico entre otras capacidades cognitivas. También, se destacan los pilares básicos de la autoestima y las estrategias de construcción del autoconcepto y se trata de relacionar estas características con las capacidades cognitivas desarrolladas mediante el aprendizaje de la programación.

El lenguaje de programación es un conjunto de símbolos y reglas semánticas y sintácticas que crean instrucciones legibles para el ordenador u otros aparatos electrónicos. Los programadores utilizan lenguajes de alto nivel para programar, y dichos lenguajes se traducen a lenguajes de bajo nivel que son los que las máquinas reconocen directamente. (Morrás, 2014)

Se podría decir que el hecho de interiorizar el funcionamiento y la lógica que siguen los aparatos tecnológicos nos permite comprender de una manera más profunda y tener una mayor seguridad y confianza a la hora de relacionarnos con ellos. Dado que la seguridad y la confianza se afianzan a medida que se

desarrollan habilidades como la resolución de problemas, la creatividad o el pensamiento crítico.

Como señala (Llorens, García, Molero, & Vendrell, 2017, p.9) ya existen estudios que relacionan las habilidades mencionadas directamente con el desarrollo del pensamiento computacional.

> En “Hacia la educación del futuro: el pensamiento computacional como mecanismo de aprendizaje generativo”, Eduardo Segredo, Gara Miranda y Coromoto León (2017), de la Universidad de La Laguna,proponen la inclusión del pensamiento computacional como un mecanismo inteligente para el desarrollo de habilidades como la resolución de problemas, el pensamiento crítico, la creatividad, la innovación y la alfabetización digital.

Por ultimo, la justificación de que Scratch 2.0 es una buena herramienta para introducir la programación en alumnos de secundaria es parte importante de este trabajo. Existen numerosos estudios que han probado que el desarrollo de proyectos por medio de Scratch 2.0 ayuda a desarrollar el pensamiento computacional, la creatividad, el razonamiento sistemático e incluso la habilidad de trabajar de manera colaborativa.

Para programar mediante Scratch no es necesario memorizar una serie de reglas semánticas. Esto hace que la programación este al alcance de un mayor número de alumnos . Se podría definir como herramienta inclusiva ya que permite a un mayor ratio de personas desarrollar la capacidad de generar contenido multimedia gracias a la programación de unos bloques predefinidos que permiten crear de una manera lógica e intuitiva las instrucciones o comandos que queremos que se ejecuten en nuestro proyecto Scratch. Existen Rampas Digitales o Assistive Technology que proporcionan adaptaciones para el Software y el Hardware de modo que la programación mediante Scratch pueda ser también una realidad para alumnos con necesidades educativas especiales, NEE.

3 OBJETIVOS

Entre los objetivos me gustaría mencionar los diferentes ámbitos y aspectos beneficiosos que aporta el hecho de aprender a programar. Así como, las capacidades cognitivas que se ven desarrolladas. También, dedico un apartado a la importancia que se brinda a las TIC en el ámbito educativo actual, justificando cuales son los objetivos que se buscan a través de ellas.

3.1 TIC

Las TIC hoy en día están consideradas como un instrumento cognitivo que facilita el aprendizaje. Las experiencias de aprendizaje que se dan a través de ellas potencia el aprendizaje por descubrimiento. Permiten desarrollar habilidades para resolver e interpretar problemas mediante la delimitación, la evaluación, interpretando y/o estableciendo relaciones entre los diferentes componentes así como, los sujetos u objetos del problema. En definitiva, evaluar e interpretar la información de la que se dispone y organizarla para obtener una solución del problema.

Si se indaga en las TIC se verá como la programación informática es una base y parte importante de las mismas. Todos los aparatos electrónicos que utilizamos en el cotidiano reciben instrucciones por medio de código de programación. Es decir, todo lo que hacen los elementos electrónicos que nos rodean viene indicado por las instrucciones de programación que se han introducido en el aparato electrónico. Es por eso, que la programación cobra importancia en las TIC y por el resto de aspectos beneficiosos que se analizarán a lo largo de la investigación que nos compete.

En conclusión, se busca un desarrollo integral donde las personas utilicen las TIC para conseguir sus objetivos y no sean meros usuarios. Que los alumnos sean capaces de utilizar las tecnologías en su doble función, es decir, ser transmisores y generadores de información y conocimiento.

Considerando la programación como parte fundamental de la integración de las TIC en el aprendizaje se destaca el motivo de que herramientas como Scratch sean un recurso didáctico apropiado para desarrollar esta competencia.

La dificultad que entraman los lenguajes tradicionales entorno a la sintaxis y semántica dificulta el acceso y uso de los sistemas computacionales a parte de la población. Existen lenguajes de programación animados como Scratch y Alice que reducen el costo del aprendizaje de reglas sintácticas y semánticas de los lenguajes de programación tradicionales, y facilitan a través del uso de elementos multimedia como imágenes y sonido, la visualización de elementos algorítmicos tales como movimiento, condiciones y repetición de acciones. (Vidal, Cabezas, Parra, & López, 2015)

Tal y como mencionan los creadores de Scratch Resnick et al. (2009), las herramientas de programación deben de tener grandes paredes, es decir, estar preparados para que personas con diferentes intereses y estilos se sientan atraídos a programar. Es por ello que Scratch ha introducido 3 características básicas en el lenguaje de programación, la programación debe ser lúdica, significativa y social.

3.2 Aprendizaje para la vida

Diferentes estudios mencionan como las habilidades que se desarrollan mediante la programación son habilidades que permiten desarrollar estrategias para tomar decisiones y resolver problemas. De esta manera, los alumnos dispondrán de una mayor cantidad de herramientas para poder tomar las mejores decisiones y poder desarrollar su futuro de la mejor manera posible.

Tal y como mencionan (Marmolejo & Campos, 2012, p 95):

> Esto no es con el fin de formar niños programadores, sino para brindarles a las nuevas generaciones las herramientas necesarias y suficientes para prepararlos para el futuro, el aprendizaje cognitivo consiste en procesos a través de los cuales el niño conoce, aprende y piensa a medida que se desarrolla, utiliza esquemas cada vez más complejos para organizar la información que recibe de su contexto y que conformará su inteligencia, así

como también su pensamiento y el conocimiento que adquiere. Fomentando la espiral del pensamiento creativo que propone (Resnick, 2007) tratando de lograr que los niños sean capaces de imaginar lo que quieren hacer, crear su proyecto de acuerdo a sus ideas, jugar con esas ideas, compartirlas con otros niños y utilizar lo aprendido para futuros proyectos, fomentando además el aprendizaje colaborativo. Marmolejo, E. (2010)

Según los creadores de Scratch la finalidad única de la aplicación no es aprender a programar sino programar para aprender a pensar de manera creativa, aprender a razonar sistemáticamente y aprender a trabajar de manera colaborativa, todas estas habilidades son consideradas esenciales para el siglo XXI.

3.3 Aprender a través de las experiencias

El aprendizaje significativo se da a través de las experiencias, se aprende haciendo y deshaciendo, es decir, experimentando. La programación nos brinda la oportunidad de crear algoritmos y tener la vivencia a través de los diferentes sentidos de comprobar cuales son las consecuencias de las instrucciones que uno mismo ha generado. El hecho de que el resultado de las órdenes que uno mismo ha creado sea una experiencia sensorial potencia la comprensión e interiorización de la información y del aprendizaje. En definitiva, se aprende haciendo, se aprende a través de las experiencias.

Las diferentes experiencias sensoriales que se dan al manejar Scratch permiten desarrollar diferentes habilidades, por ejemplo, se desarrolla la capacidad espacial ya que el escenario esta dividido en un plano cartesiano de X-Y. Al mover el ratón se muestra la información correspondiente a la posición X y Y. Esta información no es básica a la hora de dar instrucciones a los objetos aunque en ocasiones esta información es fundamental para posicionar el objeto donde uno desea.

Aprender es pues construir significados. El único conocimiento que existe es el del sujeto o sujetos que atribuyen significado a sus experiencias. (Díaz & Hernández, 2006)

3.4 Pensamiento computacional

En la introducción y justificación se han mencionado alguna de las características del pensamiento computacional. En este apartado, me gustaría mencionar con más detalle cual es la definición que han adoptado los diferentes estudios.

Tal y como menciona (Morrás, 2014, p.17,18)

> Wing (2008) define el pensamiento computacional como la habilidad de resolución de problemas, diseño de sistemas y comprensión del comportamiento humano basado en conceptos y procesos informáticos, y tiene semejanzas con el pensamiento analítico. Los dos procesos fundamentales son la abstracción y la automatización. El pensamiento computacional supone operar con múltiples niveles de abstracción simultáneamente, definiendo la relación entre los diferentes niveles, y automatizar o mecanizar esos niveles de abstracción y sus relaciones.

Se podría decir que el principio básico del pensamiento computacional consiste en conocer y reconocer el mundo de las ideas y de las representaciones, interiorizando el funcionamiento de las mismas.

3.5 Pensamiento lógico y algorítmico

El pensamiento lógico es esencial para el análisis, la comprensión y la solución de problemas. El pensamiento algorítmico implica la capacidad de definir y enunciar con claridad un problema, descomponerlo en subproblemas más manejables y mediante una serie de pasos ordenados describir una solución. Desarrollar el pensamiento lógico y algorítmico es básico para todo ser humano ya que permite analizar y dar soluciones a los problemas de la vida diaria en todos los diferentes aspectos.

Estimular el desarrollo del pensamiento lógico y algorítmico en niños y estudiantes representa una meta declarada de los sistemas educativos. Por otra parte, el uso de los lenguajes de programación potencia el pensamiento lógico y algorítmico. (Vidal et al., 2015)

3.6 Pensamiento lógico matemático

La matemática es el lenguaje de la ciencia, es un lenguaje universal común en todos los lugares del planeta. Es una técnica que permite predecir situaciones y detectar formas y patrones que nos permiten resolver problemas en nuestra vida cotidiana. Por ejemplo, cuanta cantidad de comida necesito para alimentar a X personas. Los humanos desarrollamos el pensamiento matemático como una capacidad más a la hora de crecer y adaptarnos al entorno en el que hemos crecido. La matemática ya esta reconocida como un medio universal para comunicarnos, como un lenguaje y una técnica para predecir. Se desarrolla como cualquier otra habilidad que nos permite sobrevivir en el entorno.

Según Feurzeig, Papert, & Lawler (2011) la mayoría de métodos formales de la enseñanza de las matemáticas resultan tediosos, poco motivadores y poco relacionados con el pensamiento intuitivo del alumnado. Para superar dichos obstáculos proponen una adecuada instrucción en programación informática. (Morrás, 2014)

Por otra parte, cada vez la transdisciplienaridad esta más presente en los currículum actuales, se acepta que el desarrollo computacional y el pensamiento lógico matemático están ligados como diferentes partes de un mismo todo.

Existen estudios que han decidido utilizar Scratch como herramienta para desarrollar el pensamiento lógico matemático.

4. MARCO TEÓRICO

Se relaciona el aprendizaje de la programación con teorías de aprendizaje aceptadas por la comunidad pedagógica. Se justifica porque Scratch es una herramienta didáctica apropiada para aprender a programar y que características posee para ser una herramienta inclusiva.

4.1 Constructivismo

Según Hartle, Baviskar & Smith (2012, p. 31-35)

> el constructivismo es una de las teorías del aprendizaje más aplicadas en la educación moderna. Definen el constructivismo como una teoría del aprendizaje en el que el alumno construye su conocimiento integrando, replanteando o cambiando sus estructuras cognitivas previas con las nuevas ideas o experiencias obtenidas.

Mediante la programación con Scratch se permite pasar de lo abstracto a lo concreto. Se construyen una serie de instrucciones en el ordenador por medio de bloques y la ejecución de los programas permite obtener una experiencia sensorial de lo que significan las instrucciones que se han creado. En definitiva, obtener la experiencia de que es lo que significa cada uno de los bloques y que influencia o consecuencias tiene de manera concreta.

Por consiguiente, se puede decir que es un proceso activo de aprendizaje donde el sujeto construye de manera activa y eficaz su conocimiento a partir de las experiencias vividas. Alejándose de las teorías donde el alumno es el sujeto pasivo de la transmisión del conocimiento.

4.2 Scratch

Tal y como se menciona en el estudio *Iniciación a la programación informática en educación primaria con Scratch* (Morrás, 2014, p29)

> Scratch se puede definir como un lenguaje de programación y una comunidad en línea donde se pueden crear historias interactivas, juegos, actividades, animaciones, etc., y compartir las creaciones con cualquier persona en el mundo que disponga de conexión a Internet. Al diseñar y programar proyectos de Scratch, los jóvenes y niños aprenden a pensar creativamente, razonar sistemáticamente y trabajar colaborativamente.

Se puede decir que Scratch es el sucesor de Logo, el primer programa para aprender a programar de estas características que ha existido. Scratch es un programa de código abierto que nació en el año 2003 en el laboratorio de investigaciones interdisciplinares Media Lab del Instituto Tecnológico de Massachusetts. En 2006 se publicó por primera vez Scratch 1.0 en una página web al alcance de todos los usuarios para poder descargarlo. La versión web, Scratch 2.0, vio la luz en el año 2013. El código en el que esta escrito es ActionScript, este código se utiliza en animaciones y aplicaciones Flash, gracias a eso Scratch es altamente compatible y fácilmente exportable a otras páginas Web.

Tal y como mencionan (Marmolejo & Campos, 2012, p.89).

> Scratch es una herramienta indispensable en la clase de computación de nivel básico ya que los niños deben desarrollar un pensamiento algorítmico, creatividad y destreza para resolver problemas que vayan surgiendo durante el ciclo escolar, no solo en computación sino que en cualquier materia e incluso problemas de la vida cotidiana.

Scratch se adapta a la teoría del constructivismo ya que es altamente motivador y el alumno es el sujeto activo del aprendizaje donde se cuestionan las estructuras cognitivas previas. Los alumnos podrán aprender de sus propios errores, cuando el resultado no sea el esperado tendrán que analizar lo que han hecho para encontrar el fallo y modificarlo.

La finalidad de Scratch no es preparar a las personas para carreras profesionales, es una herramienta para poder desarrollar la creatividad y el pensamiento sistemático.

4.2.1 Dr. Scratch

Dr. Scratch es una herramienta web libre de código abierto que puede resultar interesante tanto para alumnos como para profesores. Sirve para comprobar si se ha programado correctamente y recibir retroalimentación para mejorar el código. La retroalimentación, realizar pruebas, buscar errores y realizar correcciones de código con el fin de optimizarlo ofrece la posibilidad de desarrollar el pensamiento computacional.

Dr. Scratch analiza errores potenciales como código que no llega nunca ejecutarse, la repetición de código, el uso de nombres no significativos o la incialización incorrecta de atributos de objetos. Mediante este análisis se deduce la competencia demostrada del alumno en conceptos como la abstracción y descomposición de problemas, sincronización, paralelismos, pensamiento lógico o representación de la información.

Cuando Dr. Scratch detecta una puntuación baja deduce que es un novato y solo muestra la información básica para las mejoras más importantes de código. A medida que aumenta la puntuación la respuesta de Dr. Scratch también es más detallada. De esta manera los alumnos van profundizando en el aprendizaje en función de su capacidad y sus tiempos de asimilación.

4.3 Scratch como herramienta inclusiva

Hasta que aparecieron lenguajes como Scratch la programación era más tediosa, había que aprender reglas semánticas rígidas para poder programar. Los lenguajes tipo Scratch introducen una nueva modalidad para aprender a programar, un lenguaje visual adaptado para el alumnado más joven. Este lenguaje se compone por bloques que encajan unos con otros, por ello, las formas de los bloques ya nos dan pistas de las combinaciones posibles o de la estructura funcional que pueden tener los bloques. El hecho de no tener que aprender reglas de semántica complicadas hace que aprender a programar este al alcance de la mano de un mayor ratio de población.

Una de las mayores fortalezas de Scratch es su característica visual, el hecho de no tener que escribir ni una línea de código para poder crear programas, simplemente, con el puntero del ratón se arrastran los bloques con comandos predefinidos y se crea con ellos diferentes programas.

Tal y como se ha mencionado, según los creadores de Scratch (Resnick et al. 2009) esta herramienta tiene 3 características básicas:

1) Herramienta lúdica: Los bloques de colores con sus diferentes conectores están diseñados para encajar unos con otros y permiten que los alumnos puedan jugar con ellos y probar a construir programas sencillos.
2) La experiencia de programar debe ser significativa: Cuando las experiencias son personalmente significativas, el aprendizaje es más motivador. Para ello Scratch se basa en dos características, la primera, que sirva para crear cualquier tipo de proyectos, como juegos, simulaciones o animaciones y que esos proyectos se puedan personalizar mediante voces, fotos, gráficos,…
3) Se debe fomentar la interacción social. Fomentar el hecho de compartir los proyectos que cada uno crea con el fin de que otras personas lo utilicen, lo critiquen, o construyan nuevos proyectos sobre el anterior. De esta manera el aprendizaje resulta interactivo y enriquecedor.

4.3.1 Scratch herramienta para alumnos con necesidades educativas especiales (NEE)

Tal y como mencionan (López & Sanchez, 2016) en el estudio *Scratch y Necesidades Educativas Especiales: Programación para todos*, existen herramientas para adaptar el uso de Scratch a alumnos con NEE.

> Estas herramientas las encontramos en las Rampas Digitales o Assistive Technology que, aunque no son la solución ideal, sí suponen un camino de esperanza, cuando los medios estándares –teclado, ratón, pantalla o impresora en tinta- no son accesibles a las necesidades de un determinado alumno. Además en muchos casos estas adaptaciones son gratuitas.

Existen herramientas que adaptan el joystick al ratón, reconocen la voz, sustituyen el teclado por el ratón, permiten leer con los dedos o magnificar la pantalla entre otras cosas. A continuación, se muestra una tabla con algunas de las adaptaciones que existen hoy en día.

Tabla 1. Ejemplos de Rampas Digitales y Assistive Technology

Fuente: Sánchez-Montoya, 2011

Producto	Windows	GNU/ Linux Escritorio,
Adaptar teclado y ratón	Escritorio, Opciones de accesibilidad, *Emuclic*	Escritorio de Gnome
Webcam, el ojo que todo lo ve	HeadDev, CameraMouse, HeadMouse	FacialMouse
Joystick sustituye al ratón	MouseJoystick	
Reconocimiento de voz	Dragon Dictate	
Teclado sustituye al ratón	MouseKeys	
Leer con los oídos	Jaws, NVDA	Orca
Magnificar la pantalla	Zoomtext	Orca
Leer con los dedos	Línea braille, Impresoras braille	
Activar / desactivar uno o varios conmutadores		
a. Teclado virtual en pantalla	Click-N-Type	
b. Barrido intencionado por la pantalla	Kanghooru	Java, Kanghooru
c. Selección de programa por línea horizontal/ vertical.	Screen Scanner	
Diseñar Tableros de comunicación	Plaphoons	Javaplaphoons

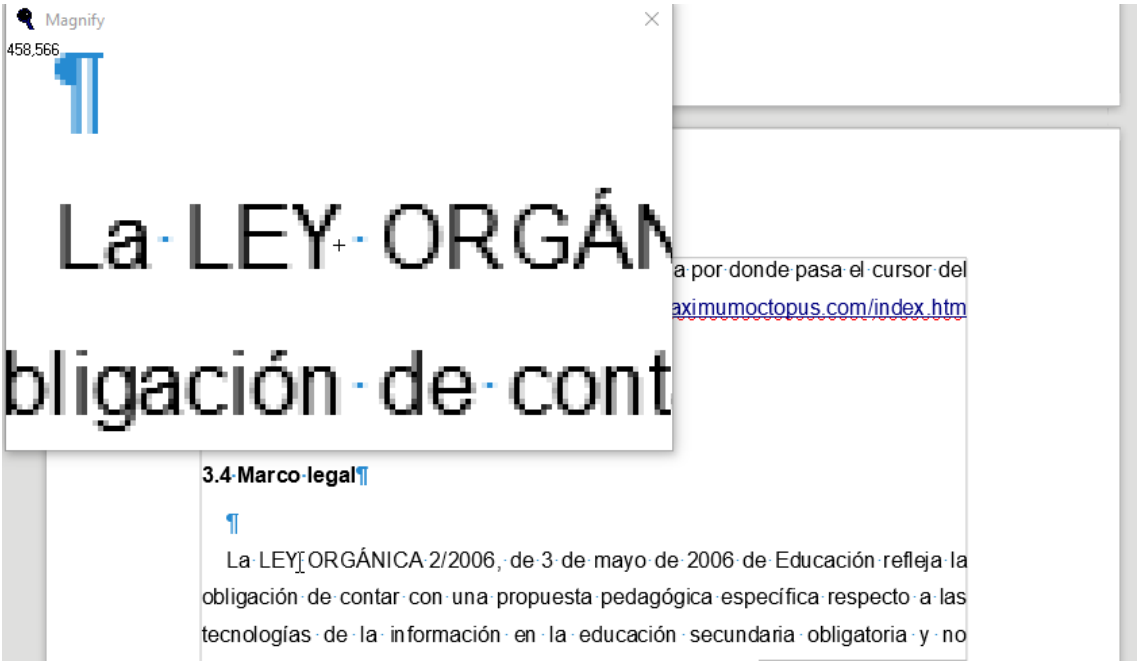
4.3.2 Ejemplos Rampas Digitales

A continuación se muestra como funciona alguna de las Rampas Digitales de la tabla 1.

Magnificar la pantalla

Este software aumenta la porción de la pantalla por donde pasa el cursor del ratón. La página del creador es http://maximumoctopus.com/index.htm (Octopus, n.d.) La descarga del software para poder realizar el ejemplos se ha realizado desde la página Educ@tic (Rodino, n.d.)

Figura 1: Magnificar pantalla

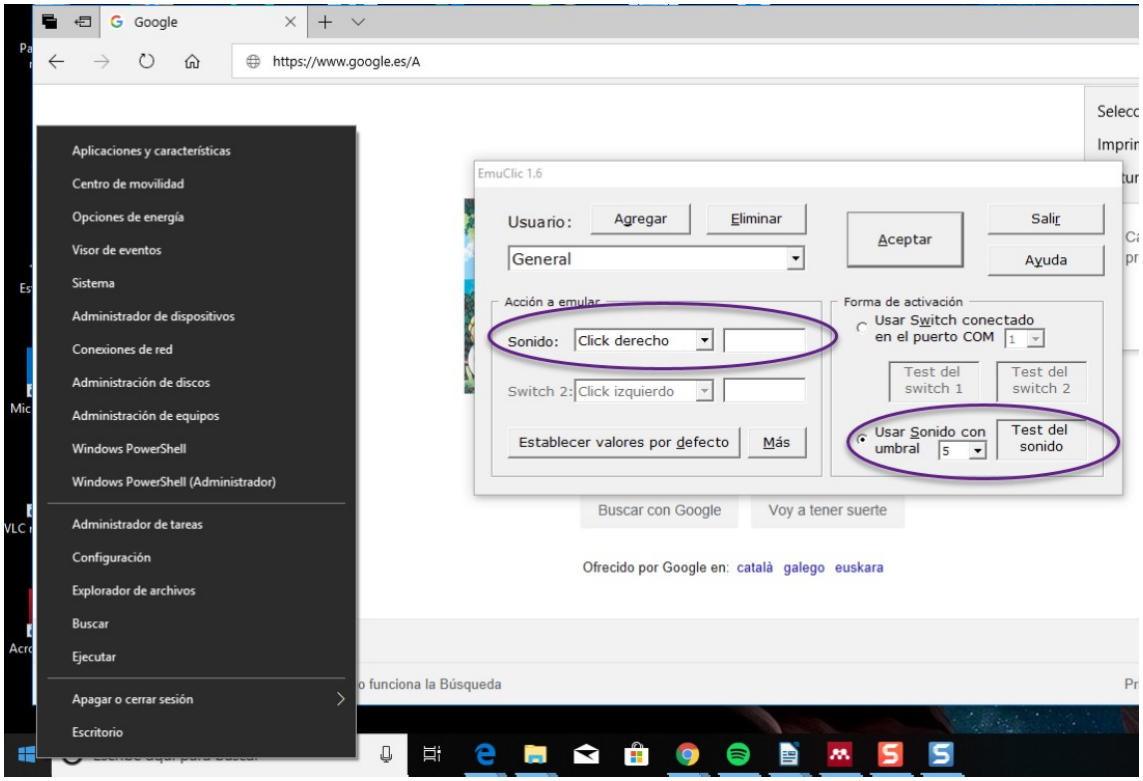


Adaptar teclado y ratón

Es una aplicación de código libre y ha sido descargada desde la página http://www.antoniosacco.com.ar/emuclic.htm El objetivo del programa es emular diversas funciones del teclado o del mouse. Cada vez que se accione el switch o se genere sonido, la computadora interpretará que se ha presionado determinada tecla o realizado un clic o etc. (Sacco, n.d.)

En el ejemplo que se muestra a continuación la aplicación emula la acción de pulsar el botón derecho del ratón cuando el sonido supera el umbral 5.

Figura 2: Adaptar teclado y ratón



4.4 Marco legal

La LEY ORGÁNICA 2/2006, de 3 de mayo de 2006 de Educación refleja la obligación de contar con una propuesta pedagógica específica respecto a las tecnologías de la información en la educación secundaria obligatoria y no obligatoria. Tanto para alumnos como para educadores. (Jefatura Del Estado, 2006, p. 17162, 17170, 17185)

> En el segundo ciclo se fomentará una primera aproximación a la lecto-escritura, a la iniciación en habilidades lógico-matemáticas, a una lengua extranjera, al uso de las tecnologías de la información y la comunicación y al conocimiento de los diferentes lenguajes artísticos.
>
> Sin perjuicio de su tratamiento específico en algunas de las materias de la etapa, la comprensión lectora, la expresión oral y escrita, la comunicación audiovisual, las tecnologías de la información y la comunicación y la educación en valores se trabajarán en todas las áreas.

> Las Administraciones educativas promoverán la utilización de las tecnologías de la información y la comunicación y la formación en lenguas extranjeras de todo el profesorado, independientemente de su especialidad, estableciendo programas específicos de formación en este ámbito. Igualmente, les corresponde fomentar programas de investigación e innovación.

Respecto a la legislación vigente en la Comunidad Autónoma Vasca, la Orden de 26 de julio de 2010, de la Consejera de Educación, Universidades e Investigación, por la que se regulan las materias optativas en Bachillerato menciona los siguientes contenidos en el apartado dedicado a las Tecnologías de la Información y la comunicación.(Gobierno Vasco, 2010, p. 34, 35)

> Bloque 3.– Introducción a la producción de software.
>
> – Programación. Algorítmica: tipos de instrucciones. Representación de diferentes procesos mediante algoritmos.
>
> – Creación de aplicaciones sencillas de software estructurado utilizando un entorno de desarrollo integrado.
>
> Bloque 4.– Internet, redes sociales y trabajo colaborativo.
>
> Bloque 5.– Elaboración y publicación de materiales multimedia.
>
> Bloque 6.– Aplicaciones de ámbito científico-técnico.

Scratch es fácilmente relacionable con las leyes educativas y también permite desarrollar trabajo complementario de diferentes áreas del currículo. Además, es una herramienta que ofrece múltiples posibilidades para desarrollar metodologías educativas actuales como la gamificación o el trabajo por proyectos.

En la ley queda reflejado claramente el fomento de las TIC, por lo tanto, el uso de Scratch queda avalado legalmente.

5. ESTADO DE LA CUESTIÓN

5.1 Docentes

Los docentes deben tener también sus propias experiencias con las TIC, aprender, analizar, indagar y reflexionar entorno a ellas. Estas reflexiones y experiencias proporcionarán habilidades para poder mejorar las experiencias educativas de los alumnos, poder escalonar los conceptos y guiar de manera correcta en el camino del aprendizaje. Por último, podrá realizar nuevas propuestas educativas con las herramientas que dispone.

Respecto al papel que deben jugar los docentes en el proceso de aprendizaje con Scratch, se puede decir, que el docente tendrá un papel instructivo en las primeras sesiones y a medida que los alumnos tengan una mayor experiencia y soltura, el rol del educador deberá cubrir las funciones de un guía y motivador.

5.2 Estudios consultados

5.2.1 Habilidades desarrolladas mediante el aprendizaje de la programación

Pea y Kurland (1984) realizan un estudio donde se analizan aportaciones al desarrollo cognitivo mediante la programación informática. Se analizan las habilidades matemáticas, la capacidad de memoria y concentración, habilidades de razonamiento analógico, habilidades de razonamiento condicional, habilidades de pensamiento procedimental y habilidades de razonamiento temporal.

El estudio de López García (2009) propone que la programación ayuda a comprender los siguientes conceptos matemáticos, concepto de variable, concepto de función, manejo de ecuaciones y gráficos y modelo matemático.

5.2.2 Estudios entorno a Scratch, Dr. Scratch y Alumnos con NEE

Entorno a las habilidades específicas que potencia Scratch existen estudios como el de Badger (2009) que afirma que Scratch aporta los siguientes conceptos de programación informática al alumnado: secuencia, iteración(ciclos), variables, sentencias condicionales, entradas vía teclado, manejo de eventos, hilos(paralelismos), números al azar, lógica booleana, diseño de interfaz de usuario, coordinación y sincronización, listas(arreglos) e interacción dinámica.

El estudio realizado por (Moreno-León, Robles, & Román-González, 2016) trata de probar la efectividad de la herramienta Dr. Scratch. Para ello, han desarrollado una serie de talleres para medir el impacto de uso en el aprendizaje.

El estudio realizado entorno a alumnos con NEE (López & Sanchez, 2016) analiza las experiencias educativas con Scratch en alumnos con discapacidad intelectual, autismo, discapacidad motriz y discapacidad visual.

5.2.3 Definición y componentes de la autoestima

El estudio realizado por (Sebastián, 2012), entorno a la autoestima y el autoconcepto docente define la autoestima como fenómeno evolutivo que se ve determinado por los diferentes factores que acompañan el crecimiento. Es decir, la autoestima se construye y se forma a través de los diferentes aspectos de la persona y la realidad interna y externa de la misma. La autoestima genera un auto reconocimiento de las capacidades de uno mismo, su ingenio y su talento, así como de sus limitaciones.

Según Franco Voli (1998) existen 5 componentes básicos de la autoestima: seguridad, identidad, integración, finalidad y competencia.

Tal y como detalla (Sebastián, 2012) solo una persona con seguridad se atreve actuar de manera confiada. Una persona segura se siente cómoda asumiendo riesgos y buscando alternativas. Esta abierta a la comunicación con los demás y es capaz de desarrollar relaciones de confianza. La integración o

la pertenencia es otro factor más que compone la autoestima, sentirse parte de un grupo, ya sea, familiar, de amigos, de trabajo, etc y y sentir que se aporta algo al grupo. Las personas con sentido de pertenencia comprenden el concepto de colaboración y participación. El principal elemento que potencia la motivación es creer que se puede conseguir la finalidad y sentir que esa meta nos satisface es otro elemento motivador fundamental. Por último, sentirse competente forma parte de la conciencia de la propia valía, la autoestima es un factor fundamental para desarrollar esta competencia.

5.2.4 Definición y construcción del autoconcepto

Shavelson, Hubner y Stanton (1976), mencionan como el autoconcepto no es más que la imagen que cada persona tiene sobre si misma, construida a través de la interpretación de su propia experiencia y del ambiente. Estas experiencias se ven afectadas de manera especial por los refuerzos y el feedback. Al autoconcepto se le concede la cualidad de poseer un sistema de organización útil para asimilar información y guiar el comportamiento, así como, tener una naturaleza dinámica para adaptarse a los requerimientos del ambiente.

Gonzales, Nuñez, Glez, y Garcia (1997) señalan que para poder adquirir la información referente a uno mismo el individuo necesita organizar las nuevas experiencias y la nueva información para poder integrarla con la información y experiencias previas. El individuo analiza y reflexiona entorno a la nueva información, si es necesario la modifica, a la vez que evalúa si le conviene o no. Este proceso beneficia de manera cualitativa y cuantitativa la estructura del autoconcepto

6 METODOLOGÍA DE LOS ESTUDIOS

Cuando se diseñan las metas que se van a proponer en los estudios entorno a los beneficios que aporta el aprender a programar hay que tener en cuenta las dificultades que muestran los alumnos al desarrollar actividades y contenidos que requieran desarrollar el pensamiento lógico para resolver problemas e intentar desarrollar una comprensión profunda de los conceptos.

Existen diferentes estudios pero la metodología de la gran mayoría de ellos trata de definir cuales son los beneficios que aporta el desarrollo de los diferentes tipos de pensamientos vinculados a la competencia digital. Para ello, se definen una serie de ejercicios donde el alumnado tendrá que desarrollar sus diferentes capacidades y los resultados se miden en función del nivel de alcance de los alumnos.

Por ejemplo, en el estudio sobre el desarrollo del pensamiento computacional con Scratch realizado en la universidad de Santiago de Chile (Barrera & Montaño, 2015) se diseñaron once sesiones de trabajo, que permitían ir desde el nivel de reconocimiento y manejo de datos hasta la automatización y la simulación de problemas de la vida diaria, de complejidad media y alta. Las sesiones están graduadas de manera jerárquica y en cada una de ellas se agregan habilidades que permiten ir alcanzando los niveles más altos del pensamiento computacional.

Los principales procesos que se detectaron en el desarrollo de las actividades fueron los siguientes, recopilar datos, analizar datos, representar datos, descomponer problemas, abstraer, algoritmos y procedimientos de automatización, simulación y paralelismo.

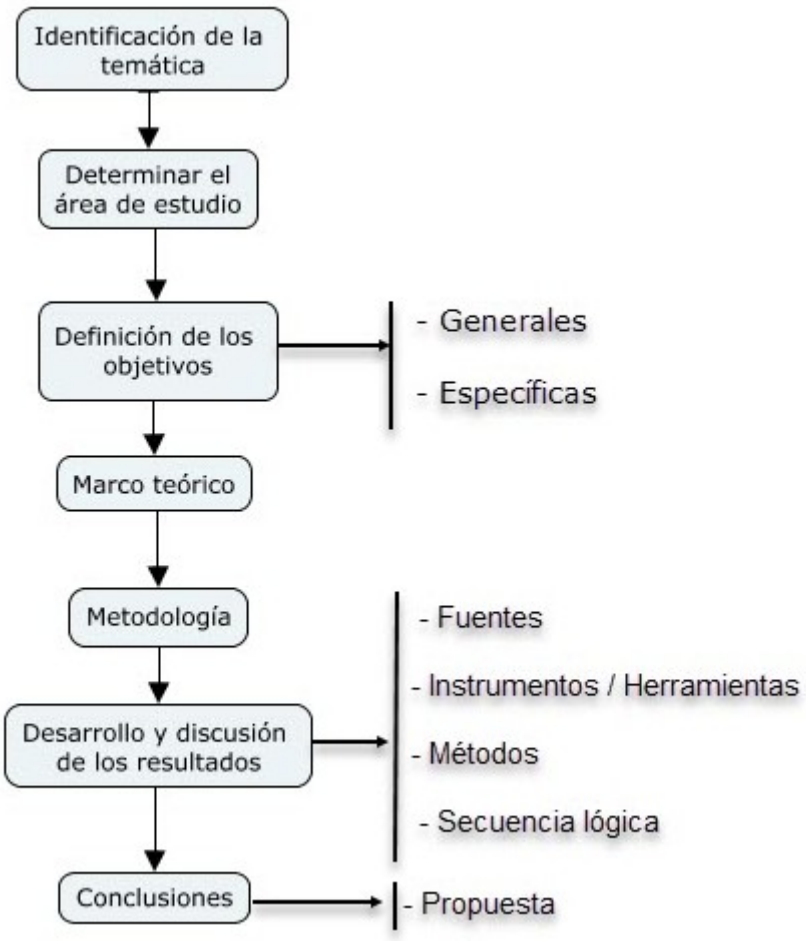
En el estudio *Dr. Scratch: Análisis Automático de Proyectos Scratch para Evaluar y Fomentar el Pensamiento Computacional* realizado para medir la efectividad de Dr. Scratch (Moreno-León et al., 2016) participaron 109 estudiantes de entre 10 y 14 años de 8 centros educativos. Los alumnos disponían de una experiencia previa programando con Scratch. Ellos analizaron uno de sus proyectos previos con Dr. Scratch, leyeron la información

del informe de resultados ofrecido por la herramienta y trataron de mejorar sus programas siguiendo las instrucciones. Finalmente, analizaron de nuevo sus proyectos.

A la hora de realizar estudios con alumnos con NEE (López & Sanchez, 2016) mencionan que se han tenido que tomar diferentes estrategias con cada uno de los casos. Existían casos donde los alumnos necesitaban rampas digitales para adaptar el Software y Hardware a sus necesidades, otros necesitaban una explicación individualizada o incluso la ayuda de un adulto para poder manejarse. Los ejercicios en general trataban de realizar un proyecto para obtener una experiencia sensorial mediante las instrucciones de programación que ellos mismos habían generado.

A continuación, se muestra una figura con la secuencia lógica que se ha seguido como metodología para esta investigación.

Figura 3: Metodología de la investigación



7 RESULTADOS

7.1 Estudios entorno a las habilidades desarrolladas mediante el aprendizaje de la programación

Existen estudios que han detectado beneficios a corto plazo y a largo plazo mediante la experiencia de aprender a programar. Por ejemplo, el estudio *La robótica Lego Mindstorms ® : un recurso didáctico para fortalecer el pensamiento lógico matemático* que analiza los beneficios de aprender a programar por medio de la robótica pedagógica. Este estudio menciona que la introducción de nuevas estrategias y recursos didácticos en el aula facilitan el proceso de aprendizaje y enseñanza, desarrollando y fortaleciendo diversas habilidades cognitivas en los estudiantes. Con el fin de obtener experiencias de aprendizaje profundas, significativas y perdurables.

Se mencionan los beneficios detectados por este estudio(Beatriz & Villamil, 2012)

Beneficios inmediatos:

• Se involucran activamente en su propio proceso de aprendizaje.

• Desarrollan la intuición científica y de ingeniería.

• Desarrollan sus intereses en matemáticas y tecnología científica.

• Potencian sus habilidades de investigación y resolución de problemas, así como lectura, escritura, habilidades de presentación y creatividad.

Beneficios a largo plazo:

• Construye auto-pensadores que además son capaces de apreciar el valor de la auto-motivación y de sentirse con recursos.

• Convertirse en un autodidacta activo.

• Fomenta la habilidad para resolver los problemas mediante estrategias centrándose en el razonamiento lógico, analítico, y pensamiento crítico. Esta habilidad es la base de muchos campos científicos así como de otras áreas profesionales.

Este mismo estudio (Beatriz & Villamil, 2012) también detectó cambios en los estudiantes, cambios por ejemplo, en su actitud y disposición a aprender, se podría decir que aumentó la motivación de los alumnos entorno al aprendizaje. Permitió a los alumnos tener una mayor comprensión de los temas de programación y en consecuencia, de igual modo, aumentó su capacidad de ser más autónomos en el aprendizaje. Los proyectos colaborativos donde se puede compartir con personas a lo largo de todo el planeta promueven el desarrollo de la capacidad colaborativa. También, se evidencia un beneficio de las habilidades motrices, las habilidades de concentración, observación y comunicación y en el desarrollo del pensamiento lógico matemático.

Por último, también se detectaron cambios en los docentes, por ejemplo, el hecho de tener nuevos sistemas de seguimiento para los procesos de enseñanza y aprendizaje o la creación de nuevas formas de evaluar a los estudiantes. Se apreció una mayor motivación y entusiasmo por parte de los docentes y una mayor predisposición para utilizar recursos didácticos innovadores.

7.1.1 Beneficios del desarrollo del pensamiento computacional y el pensamiento algorítmico.

Los resultados del estudio *Desarrollo del Pensamiento Computacional con Scratch* de (Barrera & Montaño, 2015) muestran como el estudiante puede llegar a tener un pensamiento computacional avanzado con actividades diseñadas en forma lúdica y entretenida.

También, se destaca el hecho de que las actividades interactivas para generar pensamiento computacional de alto nivel son una herramienta efectiva que ayuda a fortalecer los procesos lógicos que permiten la modelación correcta para la solución de problemas, además, de fomentar habilidades como la creatividad. Por último, se menciona como el hecho de poder compartir el trabajo realizado resulta gratificante para el alumnado, el hecho de que otra persona lo pueda utilizar supone un desafío para tratar de realizar el trabajo lo más atractivo posible.

La prueba y el error es una estrategia para desarrollar el pensamiento algorítmico. Mediante el estudio *Experiencias Prácticas con el Uso del Lenguaje de Programación Scratch para Desarrollar el Pensamiento Algorítmico de Estudiantes en Chile* realizado por (Vidal et al., 2015) se demuestra que fue posible comprobar que las alternativas de respuesta a una pregunta lógica, generan un razonamiento en el estudiante sobre cuál es la opción correcta así cómo validar sus soluciones.

En el estudio *Desarrollo del pensamiento computacional en educación primaria: una experiencia educativa con scratch* se analiza directamente el desarrollo del pensamiento computacional y los resultados positivos permiten confirmar que Scratch favorece la adquisición de conceptos computacionales.

Este estudio concluye diciendo que(Álvarez, 2017, p. 62):

> la afirmación de que utilizar la herramienta Scratch no solo implica mejorar las habilidades propias de la programación computacional, como es el Pensamiento Computacional, si no que se trata de una herramienta que potencia contenidos transversales tales como el razonamiento lógico-matemático, la autonomía personal, la creatividad y el trabajo en grupo.

Por último, el estudio *Visual programming languages integrated across the curriculum in elementary school: A two year case study using "Scratch" in five schools* concluye que observando los resultados del antes y el después se encontraron mejoras significativas en cuanto a la creatividad computacional. Por lo tanto, se puede decir que el uso de la herramienta Scratch mejora capacidades como la lógica, el aprendizaje de conceptos de programación y la práctica computacional (Sáez-López, Román-González, & Vázquez-Cano, 2016).

7.2 Estudios entorno a Scratch, Dr. Scratch y alumnos con NEE

El estudio *Evaluating Scratch to introduce younger schoolchildren to programming* de Wilson y Moffat (2010) afirma que tras aprender a manejar Scratch durante 8 sesiones formativas se produjo un cambio en el plano

afectivo ya que se pudo observar que la mayoría del alumnado se sentía más cómodo en el aula y demostraba una mayor motivación.

En el estudios *Programar con Scratch en contextos educativos: ¿Asimilar directrices o co-construir Tecnologías para la Inclusión Social?* Se analiza Scratch como parte de las Tecnologías para la Inclusión Social (TIS). El estudio se centra en los siguientes aspectos. El desarrollo del pensamiento computacional como eje de las prácticas, y la creación de comunidades en torno a la herramienta y diferentes tipos de participación en torno a la misma. (Monjelat & San Martín, 2016). Los resultados muestran como las limitaciones de Scratch no están vinculadas a las barreras para la co-construcción de Tecnologías para la Inclusión Social. Estas limitaciones vienen dadas por la orientación instrumental que tienen las practicas de aprendizaje y enseñanza que se han realizado. Este estudio marca el desafío de la innovación en las tecnologías informáticas para la inclusión social. Avanzar para transcender la individualidad y considerar aspectos prioritarios de la problemática social. Realizando procesos de construcción en un marco de alianzas socio-técnicas.

El estudio entorno a Dr. Scratch (Moreno-León et al., 2016) determina que esta herramienta es atractiva para la mayoría de los alumnos, estos consideran que es fácil analizar proyectos y se sintieron bien cuando la herramienta les mostró los resultados del análisis y su puntuación relacionada con el pensamiento computacional. Unicamente un 3% declaró sentirse mal al recibir la puntuación. La gran mayoría de los alumnos fueron capaces de entender la respuesta de Dr. Scratch pero un 5,7% indicó que no comprendió la información recibida. Se puede considerar que Dr. Scratch promueve el aprendizaje y por lo tanto, la motivación y el deseo de querer mejorar las habilidades de programación.

Los autores del estudio con Dr. Scratch (Moreno-León et al., 2016) afirman que hay diferencias significativas entre las pruebas pre y post, lo que indica que el uso de Dr. Scratch ayudó a los alumnos participantes a desarrollar su pensamiento computacional. También, se menciona que la herramienta parece menos útil para aquellos alumnos con una puntuación inicial alta, donde los incrementos son más difíciles de conseguir.

El resultado del estudio *Scratch y Necesidades Educativas Especiales: Programación para todos* (López & Sanchez, 2016, p. 12) menciona lo siguiente.

> al utilizar Scratch con alumnos con NEE los estudiantes se sienten protagonistas de su proceso de aprendizaje, esto les motiva enormemente y **favorece su autoconcepto**. Es increíble como estudiantes que pensábamos que no podían hacer nada o muy poco, son capaces de planificar, establecer hipótesis y de realizar preguntas que no surgirían en cualquier otro contexto de aprendizaje.

Informes como el de (Delors, 2006) abalan el hecho de que aprender a programar con Scratch cubre los 4 pilares de la educación del siglo XXI, estos son, aprender a conocer, aprender a hacer, aprender a vivir juntos y aprender a ser. El estudiante debe poner en práctica estos pilares a la hora de dar respuesta a el interrogante y a la hora de generar el algoritmo.

8. CONCLUSIONES

El desarrollo de las competencias computacionales mediante el aprendizaje de la programación desarrolla ciertas competencias que permiten procesar y gestionar abundante información compleja de manera más estructurada. Esto es una herramienta que ayuda a tomar decisiones para resolver problemas reales. Otra de las competencias que se desarrolla gracias a el uso de las TIC es el hecho de aprender a trabajar en entornos colaborativos formales e informales, con el fin de generar producciones responsables y creativas para compartirlas por medio del entorno digital.

En consecuencia, la competencia digital trata de sacar el máximo partido a las tecnologías de la información mediante la comprensión profunda de las mismas, es decir, la comprensión de su naturaleza, su manera de funcionar y los cambios que suponen en el ambiente social y laboral. Se desarrolla un pensamiento crítico entorno a el gran flujo de información y se es capaz de analizar la información de manera colaborativa o de manera autónoma. Utilizar las herramientas tecnológicas para organizar y procesar la información con el fin de orientarla hacia los objetivos y fines previamente establecidos. Objetivos y fines entorno al aprendizaje, al trabajo o al ocio. En definitiva, se trata de utilizar los medios tecnológicos para tratar de dar respuesta a problemas reales de modo eficiente.

Diferentes estudios demuestran que las herramientas como Scratch 2.0 sirven para desarrollar el pensamiento computacional, algorítmico, lógico y matemático. Así como la colaboración y el trabajo en equipo. También, es una herramienta que permite tener una experiencia sensorial del resultado de los algoritmos y incorpora el aprendizaje por medio del descubrimiento, de modo que, es una herramienta motivadora para el aprendizaje. Me gustaría remarcar como las competencias digitales y computacionales van más allá del mero hecho de utilizar las diferentes herramientas tecnológicas. Cuando se evalúan trabajos en aplicaciones como Scratch no solo se debe medir si se ha obtenido la resolución del problema, sino que se deben proponer criterios de calificación

que tengan en cuenta los aspectos como, el proceso, el funcionamiento, la interfaz gráfica, la creatividad, la programación, el pensamiento computacional y si se comparte con la comunidad educativa o no. El alumnado con necesidades educativas específicas deberá ser evaluado teniendo en cuenta sus necesidades de manera más personalizada.

Me gustaría destacar los beneficios y las habilidades entorno al aprendizaje de la programación y el pensamiento computacional que me parecen más significativas y que más veces se mencionan en los diferentes estudios. Por ejemplo, la habilidad para resolver los problemas por medio de estrategias centrándose en el razonamiento lógico, analítico, y pensamiento crítico. Así como, fortalecer los procesos lógicos que permiten la modelación correcta para la solución de problemas. Al igual que, el desarrollo de las habilidades motrices, las habilidades de concentración, observación, comunicación y creatividad.

Las herramientas como Scratch promueven el aprendizaje, aumentan la motivación y desarrollan la capacidad de ser más autónomos. Este tipo de herramientas promueven el desarrollo de la capacidad colaborativa ya que compartir el trabajo realizado resulta gratificante .

Me gustaría apreciar como ciertas características y habilidades mencionadas entorno al desarrollo del autoconcepto y autoestima son capacidades que también se desarrollan mediante el aprendizaje de la programación y el pensamiento computacional. Incluso con el aprendizaje de Scratch dado que esta herramienta fomenta el hecho de compartir con la comunidad, el trabajo en grupo, la colaboración y la participación. Por lo tanto, características básicas de la autoestima como la seguridad, la identidad, la integración y la finalidad, o pilares del autoconcepto, como la capacidad de interpretar las experiencias y el contexto requieren de sistemas de organización útiles para asimilar información y guiar el comportamiento. Es decir, existe la necesidad de organizar las nuevas experiencias y la nueva información para poder integrarla con la información y experiencias previas y poder reflexionar entorno a ella.

De igual modo, me gustaría destacar el hecho de que una comprensión profunda entorno al funcionamiento de las herramientas tecnológicas aporta una mayor seguridad a la hora de relacionarse con ellas. Por consiguiente, se

puede decir que el alumno dispondrá de una mayor seguridad en el mismo en las relaciones que establezca con las herramientas tecnológicas.

Como propuesta de investigación futura quiero presentar el siguiente cuestionario para poder dárselo a los alumnos que hayan adquirido unos conocimientos básicos de programación mediante Scratch 2.0 o una herramienta similar. Las preguntas giran entorno a conceptos como la motivación, la seguridad, las estrategias para resolver problemas, la comunicación, la autonomía y la capacidad colaborativa. En definitiva, capacidades desarrolladas por medio de la competencia digital y el aprendizaje de la programación y a su vez también, conceptos que forman parte de la autoestima y el autoconcepto.

Tabla 2 : Cuestionario para alumnos con nociones de programación que hayan desarrollado su pensamiento computacional

	Muy de acuerdo	Algo de acuerdo	Ni de acuerdo ni desacuerdo	Algo en desacuerdo	Muy desacuerdo
¿ Consideras que tienes una mayor comprensión entorno al funcionamiento de la tecnología ?					
¿ Te sientes más capaz de solucionar problemas relacionados con el Software ?					
¿ Te sientes más capaz de solucionar problemas relacionados con el Hardware ?					
¿ Consideras que tienes una mayor capacidad para organizar la información ?					
¿ Crees que este aprendizaje influye en la manera en la que te relacionas con la tecnología ?					
¿ Te sientes más seguro de ti mismo a la hora de manejar herramientas tecnológicas ?					
¿Te gusta compartir tus proyectos con los demás?					
¿Te gustaría realizar más proyectos utilizando la programación?					

9. REFERENCIAS

Álvarez, M. (2017). Desarrollo del pensamiento computacional en educación primaria : una experiencia educativa con scratch. *Revista de Ciencias de La Educación*, 45–64.

Badger, M. (2009). Scratch 1.4: Beginner's Guide. Packt Publishing Ltd.

Barrera, R., & Montaño, R. (2015). Desarrollo del Pensamiento Computacional con Scratch. *Nuevas Ideas En Informática Educativa TISE*, 616–620.

Beatriz, L., & Villamil, L. (2012). La robótica Lego Mindstorms ® : un recurso did áctico para fortalecer el pensamiento lógico matemático. *Perspectivas Docentes Acotaciones*, 12. Retrieved from http://revistas.ujat.mx/index.php/perspectivas/article/viewFile/561/467

Delors, J., La Educación Encierra un Tesoro, Santillana, Ediciones Unesco, (2006).

Díaz, Frida; Hernández, Gerardo (2006). Estrategias docentes para un aprendizaje significativo. México, D.F.: McGraw Hill.

Feurzeig, W., Papert, S. A. & Lawler, B. (2011). Programming-languages as a conceptual framework for teaching mathematics. Interactive Learning Environments, 19(5), 487- 501. doi: 10.1080/10494820903520040

Gobierno Vasco. (2010). ORDEN de 26 de julio de 2010, de la Consejera de Ed ucación, Universidades e Investigación, por la que se regula la ordenación y el proceso de evaluación en el Bachillerato., 15872–15877.

GONZALES, P. J.; NUÑEZ, J.; GLEZ,S.; & GARCIA, M. (1997) Autoconcepto, autoestima y aprendizaje escolar. En Revista Psicothema, Vol. 9, número 002. Universidad de Oviedo. España.

Hartle, R. T., Baviskar, S. & Smith, R. (2012). A Field Guide to Constructivism in the College Science Classroom: Four Essential Criteria and a Guide to Their Usage. Bioscene: Journal of College Biology Teaching, 38(2), 31-35.

Jefatura Del Estado. (2006). Ley Orgánica 2/2006 de 3 de Mayo, 17158–17207.

Llorens, F., García, F. J., Molero, X., & Vendrell, E. (2017). La enseñanza de la informática, la programación y el pensamiento computacional en los estudios preuniversitarios. *Education in the Knowledge Society (EKS)*, *18*(2), 7. https://doi.org/10.14201/eks2017182717

López, C., & Sanchez, R. (2016). Kids Code in a rural village in Norway: could code clubs be a new arena for increasing girls' digital interest and competence? *Information Communication and Society*, *19*(1), 95–110. https://doi.org/10.1080/1369118X.2015.1093529

López García, J. C. (2009). Algoritmos y Programación: Guía para docentes. Recuperado de http://www.eduteka.org/pdfdir/AlgoritmosProgramacion.pdf

Marmolejo, E. (2010) Introducción a la programación infantil para desarrollar la creatividad en niños de primer y se- gundo grado de primaria utilizando scratch. Memorias SOMECE 2010.

Marmolejo, J. E., & Campos, V. (2012). Pensamiento lógico matemático con scratch en nivel básico logical mathematical thinking with scratch in basic level, 87–95.

Monjelat, N., & San Martín, P. S. (2016). Programar con Scratch en contextos educativos: ¿Asimilar directrices o co-construir Tecnologías para la Inclusión Social? *Praxis Educativa*, *20*(1), 61–71.

Moreno-León, J., Robles, G., & Román-González, M. (2016). Dr. Scratch: Análisis Automático de Proyectos Scratch para Evaluar y Fomentar el Pensamiento Computacional. *Revista de Educación a Distancia (RED)*, *46*(46). https://doi.org/10.6018/red/45/10

Morrás, A. (2014). INFORMÁTICA EDUCATIVA Hasier MORRÁS ARANOA.

Octopus, M. (n.d.). Welcome :: MaximumOctopus. Retrieved June 24, 2019, from http://maximumoctopus.com/index.htm

Pea, R. D. & Kurland, D. M. (1984). On the cognitive effects of learning computer programming. New ideas in psychology, 2(2), 137-168. Resta,

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E.,

Brennan, K., Millner, A., Rosenbaum, E., Siver, J., Silverman, B., y Kafay, Y. (2009). Scratch: Programming for all. Communications of the ACM, 52 (1), 60-67.

Resnick, P. M. (2007). Las nuevas tecnologías ayudan a los estudiantes a navegar la espiral del pensamiento creativo, *5191*, 1–5. Retrieved from https://web.media.mit.edu/~mres/papers/sowing-seeds-spanish-translation.pdf

Rodino, M. E. (n.d.). Descargar rampas - AplicaTics. Retrieved June 24, 2019, from, https://sites.google.com/site/aplicatics/4-rampas-digitales/descargar-rampas

Sacco, A. (n.d.). Antonio Sacco - Software EmuClic. Retrieved June 24, 2019, from http://www.antoniosacco.com.ar/emuclic.htm

Sáez-López, J.-M., Román-González, M., & Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using “Scratch” in five schools. *Computers & Education*, *97*, 129–141. https://doi.org/10.1016/J.COMPEDU.2016.03.003

Sánchez-Montoya, R. (2011). ¿Más avance tecnológico implica mayor inclusión? VII Jornadas de Cooperación Educativa con Iberoamérica sobre Educación Especial e Inclusión Educativa. Octubre, 2011, Montevideo, Uruguay.

Sebastián, V. H. (2012). Autoestima y Autoconcepto docente/ Self-esteem and Teacher Self-concept. *Phainomenon*, *11*(1), 23–33. Retrieved from http://www.unife.edu.pe/publicaciones/revistas/filosofia/Phainomenon/2012/articulo 2.pdf

Shavelson, R. J., Hubner, J. J. y Stanton, G.C (1976). Self-concept: Validation of construct interpretations. Review of Educational Reseach.

Segredo, E., Miranda, G., & León, C. (2017). Hacia la educación del futuro: el pensamiento computacional como mecanismo de aprendizaje generativo. Education in the Knowledge Society, 18(2).

Vidal, C. L., Cabezas, C., Parra, J. H., & López, L. P. (2015). Experiencias prácticas con el uso del lenguaje de programación scratch para desarrollar el pensamiento algorítmico de estudiantes en Chile. *Formacion Universitaria*, *8*(4), 23–32. https://doi.org/10.4067/S0718-50062015000400004

VOLI, F. (1998). “La autoestima del profesor, manual de reflexión y acción educativa”. PPC editorial. Madrid

Wilson, A., & Moffat, D. (2010). Evaluating Scratch to introduce younger schoolchildren to programming. In Psychology of Programming Interest Group 2010 Workshop.Recuperado de http://scratched.gse.har-vard.edu/sites/default/files/wilson-moffat-ppig2010-final.pdf

Wing, J. M. (2008). Computational thinking and thinking about computing. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 366(1881), 3717-3725.

Wing, J. M. (2011, March). Computational thinking. Recuperado de: https://pdfs.semanticscholar.org/f0fe/32388ff4472e92c17753e8689ac56ff85bc1.pdf